DevSecOps Project — Initial Planning Document Submission

Team Details

Team Number: 4

Team Name: SafeStack

Member Names & Roll Numbers:
- Nikhil Kalloli (sec-B) — 1MS22CS091
- Nitin Paramker (sec-B) — 1MS22CS095
- Sahil Sahay — 1MS22CS119
- Saurabh Kushwaha — 1MS22CS126

Role of each member:
- Sahil Sahay — CI/CD pipeline and integration
- Nikhil Kalloli — Containerization, Kubernetes setup, and environment isolation
- Nitin Paramker — Real-time monitoring features and security enforcement
- Saurabh Kushwaha — Faculty dashboard UI and logging system

Project Title and Description

Title of the application: Secure Lab Execution & Monitoring System

Brief description:
A secure and monitored web-based coding environment for college lab sessions. The system allows students to access a browser-embedded IDE while restricting unethical behavior such as tab switching or external copy-pasting. Faculty can monitor all activities in real-time.

Objective

The goal of this project is to build a fully functional DevSecOps pipeline that supports a secure execution environment using containerized labs. The pipeline will automate the provisioning, deployment, and monitoring of individual student containers and ensure logs and activities are traceable. This helps promote ethical coding practices and simplifies faculty supervision.

Tools & Technologies (Initial Selection)

IaC: Terraform (for infrastructure automation), Ansible (for configuration management)

CI/CD: GitHub Actions (for version control integration and deployment workflows)

Containerization: Docker (per-student isolated containers)

Testing: Postman (API testing), custom scripts for simulation and tab-switch scenarios

Monitoring: WebSockets (for real-time logs), custom dashboard using ReactJS

Secrets/Security: Secure login (ID + session code), container isolation, browser policy control

Initial Timeline Planned Activities

- Week 1: Team formation, finalize idea, assign roles
- Week 2: Set up infrastructure using Docker and Terraform; prepare lab templates (Python, C, NS3)
- Week 3: Integrate frontend (ReactJS) with IDE (xterm.js/Monaco) and backend (Node/Flask)
- Week 4: Implement real-time log streaming via WebSockets and security restrictions like tab tracking
- Week 5: Faculty dashboard for activity monitoring and container session management
- Week 6: Conduct testing (unit, manual, and load tests), prepare for deployment

Expected Risks or Challenges

- Ensuring secure isolation of each container session for students
- Handling accurate tab-switch and clipboard detection at the browser level
- Real-time logging and monitoring under high concurrency (30–100 students)
- WebSocket reliability and container failure recovery mechanisms